

AMENDMENTS TO THE SPECIFICATION

Please amend the specification of the present application as set forth below. Changes to the specification are shown by strikethrough (for deleted matter) and underlining (for added matter).

Please amend paragraph [0019] as follows:

-- Figure 1 shows an environment 100 in which a computing device can execute a first software program so as to emulate the operation of a low resource client through the use of emulated memory. Figure 1 shows the computing device at reference numeral 102, the first software program as being one or more emulator applications (Apps) 126. Computing device 102 is in communication via a network interface 112 to a communications network ~~106~~104. A user input device, such as a mouse 108, provides input to software executing in hardware that is located in a tower or case 106. --

Please amend paragraph [0025] as follows:

-- Each emulator application 126, when executed by processing unit 110, emulates the execution of an active application 130, such as on a low resource client. Memory manager 134 is called by the active application 130 to access one or more previously allocated contiguous portions (*n*) 140 of emulated memory 132 by performing an emulated memory access operation. The emulated memory access operation can be an attempt to read from or write to one ~~or~~for more previously allocated contiguous portions (*n*) 140. Alternatively, the emulated

1 memory access operation can be an attempt to reallocate or to free one ~~or~~for more
2 previously allocated contiguous portions (n) 140. --
3

4 Please amend paragraph [0029] as follows:
5

6 -- Memory Allocation.

7 Prior to process 200, an allocation operation allocated one or more
8 contiguous portions (n) 140 of emulated memory 132. The allocation operation
9 assigned an ID or generation count to all addresses within the contiguous portions
10 (n) 140 of emulated memory 132. The assigned ID or generation count is entered
11 into one or more entries (j) 122 in the table 120 that maps to the emulated memory
12 132. The table 120 can be indexed by the index 'j' representing the addresses of
13 emulated memory 132. As such, each table entry (j) 122 can correspond to an
14 address of a contiguous portion (n) ~~144~~140 of emulated memory 132. As part of
15 the allocation operation, the assigned ID or generation count is copied from a
16 corresponding table entry (j) 122 into unused bits in a pointer (i) 136 to produce
17 pointer (i) 138 that points to an address of the allocated contiguous portion (n) 140
18 of emulated memory 132. Thus, each pointer (i) 138 to emulated memory 132 and
19 each table entry (j) 122 in the table 120 contains an ID or a generation count.
20 Thereafter, pointer (i) 138 can be used by a call by active application 130 to the
21 memory manager 134 to perform an emulated memory access operation upon the
22 previously allocated contiguous portions (n) 140 in emulated memory 132. --
23
24
25

1 Please amend paragraph [0031] as follows:
2

3 -- At block 206, the table entry (j) 122 is retrieved from table 120 that
4 corresponds to the address of the one or more previously allocated contiguous
5 portions (n) 140 of emulated memory 132. The emulator application 126 performs
6 a comparison to see if there is match between the ID or generation count that is
7 contained in the pointer (i) 138 and retrieved the table entry 122. When the ID or
8 the generation count is the same, block 208 passes control of process 200 to block
9 212. Otherwise, a diagnostic is output at block 210. The diagnostic can be
10 descriptive of the circumstances of the error, such as where there has been an
11 attempt by the active application 130 to call the memory manager 134 for a
12 memory access operation to a contiguous portion (n) 140 using an obsolete
13 memory pointer in an incorrect access to emulated memory 132. When such an
14 incorrect emulated memory access is attempted during the execution of active
15 application 130, the execution can be stopped so that a developer can locate and
16 debug the code. As a result, the developer need not undertake the task of finding
17 how and where the bug occurred, should execution of the active application 130
18 have been permitted to continue beyond the occurrence of the incorrect access.
19 Alternatively, the diagnostic can be descriptive enough to allow debugging even
20 though the active application 130 is permitted to continue execution after the
21 occurrence of the incorrect access to emulated memory. --
22
23
24
25